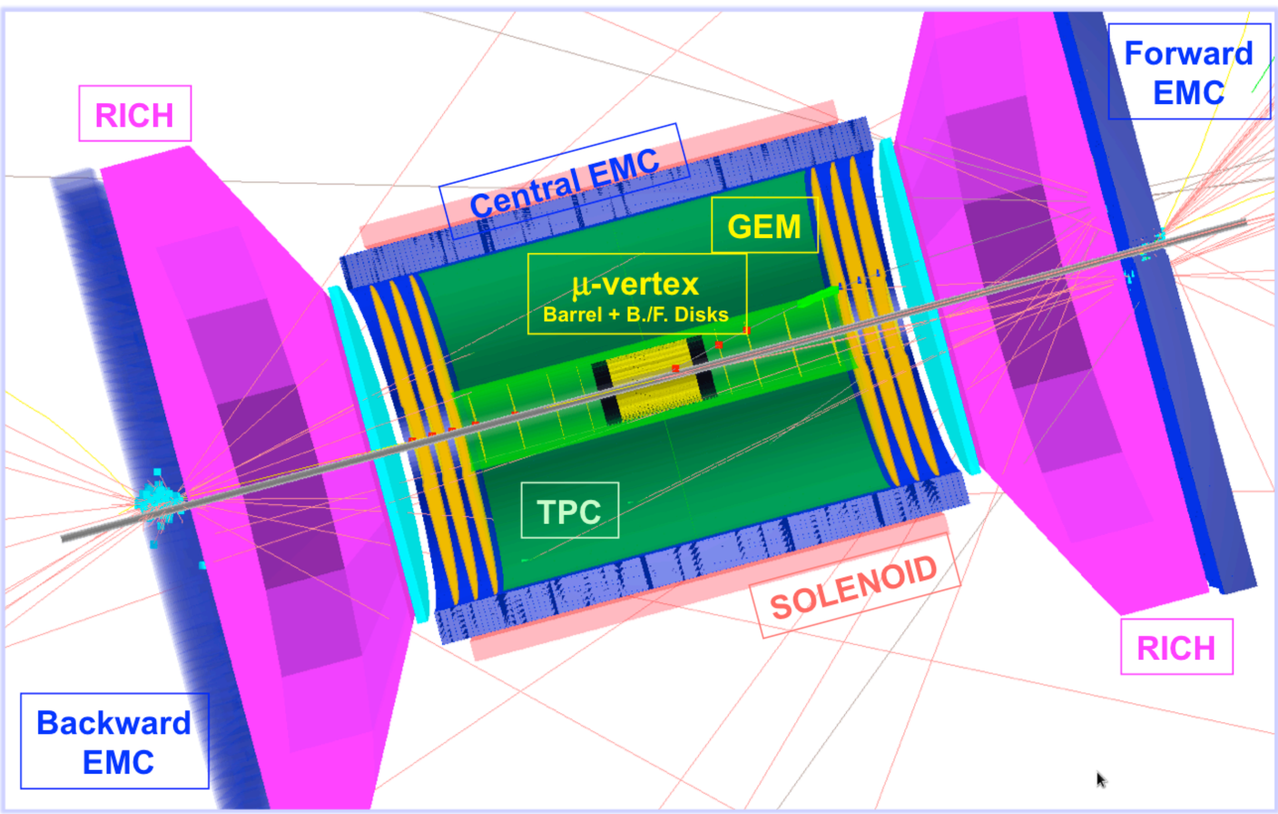
November 2013

The eRHIC Software Overview



The BNL EIC Science Taskforce

Elke-Caroline Aschenauer, Mark Baker, Thomas Burton, Benedetto Di Ruzza, Kjeld Oleg Eyser, Salvatore Fazio, Alexander Kiselev, Matt Lamont, J.-H. Lee, Marco Stratmann, Tobias Toll, Thomas Ullrich and Liang Zheng

Table of Content

1 Introduction

This document gives a brief overview of the software packages that are currently in use in the EIC Task Force at BNL. Some of the programs and packages were adopted from existing ones and modified to match our needs; others were developed from scratch where no suitable software was available. We describe their purpose, how they are maintained and made available for a broader community, and how they are archived. These programs are vital for developing the physics case for an EIC and form the foundation for any further studies towards the realization of an EIC throughout the NSAC Long Range Plan process and beyond.

1 General Software Concept:

All the EIC/eRHIC software is available on /afs/rhic.bnl.gov/eic/. The general philosophy in developing software was to base the code on community-wide, open source software, as this guarantees good documentation and long-term support, e.g. the CERN libraries, ROOT and other tools.

All software has been developed under Linux, and a computing platform of 10 multicore nodes, integrated into RACF, has been used. The EIC nodes are fully integrated into the RACF, and follow the same software upgrade schedule as the rest of RACF. RACF provides some software, such as the CERN libraries, while others, like ROOT and LHAPDF, use EIC-specific installations.

Version control of EIC-developed software is generally performed using Subversion (svn), while third-party codes are maintained by their developers. An ITD-provided repository at http://svn.racf.bnl.gov/svn/eic/ hosts the EIC software. This repository is publicly readable, but has write privileges restricted to EIC developers. Currently a portion of, though not all, EIC-developed software is committed to this repository.

All Monte Carlo (MC) analysis is performed within ROOT, an object-oriented analysis framework written in C++. Most MC event generators (section 1.1) used in the group are written in Fortran and do not easily interface with ROOT. Rather than writing code to allow each generator to interact with ROOT directly, events are instead outputted in ASCII format according to a standard format. Each event is written with a "header" line containing event-wide information, followed by lines describing the particles in that event, one particle per line. The particle information follows a shared HEP-inspired format, while the header line is generator-specific. These ASCII files are then parsed to build C++ events objects that are stored in ROOT file in a tree format. The code to build these trees is distributed as part of the eic-smear package (section 1.2).

1.1 Supported MC Generators:

In the following we describe in detail the physics generators developed, modified or frequently used to make predictions for the EIC physics potential. They are listed in alphabetical order. Each of the generators has detailed documentation at the EIC wiki page https://wiki.bnl.gov/eic/index.php/Simulations

1.1.1 Electron-Proton Generators

Package Name: DJANGOH
Purpose/Usage: Simulates deep inelastic (un)polarized lepton-proton scattering for both NC and CC events, including both QED and QCD radiative effects.
Language(s): Fortran
Version Control: SVN and http://wwwthep.physik.uni-mainz.de/~hspiesb/djangoh/djangoh.html
Location: http://wwwthep.physik.uni-mainz.de/~hspiesb/djangoh/djangoh.html
and locally at /afs/rhic.bnl.gov/eic/PACKAGES/DJANGOH-4.6.10
Platforms: Linux, Mac OS X
Maintainers: H. Spiessberger, University of Mainz, Germany
Documentation: Extensive documentation is distributed with the package.
Future Updates/Additions Planned: Include polarization effects for NLO NC processes and include (un)polarized semi-inclusive CC events
Comments:

Package Name: GMC_TRANS
Purpose/Usage: A Monte Carlo generator for semi-inclusive deep inelastic scattering (SIDIS). It simulates single hadron production from the scattering of a lepton on a transversely polarized hadron. It includes the hadron transverse spin ("transversity") distribution and transverse-momentum-dependent (TMD) distributions, such as the Sivers distribution function.
Language(s): Fortran and C
Version Control: SVN
Location: /afs/rhic.bnl.gov/eic/PACKAGES/gmc_trans
Platforms: Linux, Mac OS X
Maintainers: Tom Burton
Documentation: https://wiki.bnl.gov/eic/index.php/Gmc_trans
Future Updates/Additions Planned: None at this moment
Comments:

Package Name: LEPTO
Purpose/Usage: A Monte Carlo generator for deep inelastic scattering; it is the basis for DJANGOH, PEPSI and LEPTO_PHI
Language(s): Fortran
Version Control: SVN
Location: http://home.thep.lu.se/~rathsman/lepto/ and locally at /afs/rhic.bnl.gov/eic/PACKAGES/lepto
Platforms: Linux
Maintainers: Prof. G. Ingelmann at University of Uppsala and Prof. J. Rathsman at University of Lund
Documentation: the manual is distributed with the tar-ball of the code and there is documentation at https://wiki.bnl.gov/eic/index.php/LEPTO
Future Updates/Additions Planned: None at this moment
Comments:

Package Name: LEPTO_PHI
Purpose/Usage: LEPTO-PHI is a version of LEPTO, which has been modified to include azimuthal asymmetry effects arising at the cross-section level in unpolarized (or longitudinally polarized) DIS. These effects are included to all orders in k_t/Q and up to $O(\alpha_s)$.
Language(s): Fortran
Version Control: SVN
Location: locally at /afs/rhic.bnl.gov/eic/PACKAGES/lepto_phi
Platforms: Linux
Maintainers: Mark Baker
Documentation: There is documentation at https://wiki.bnl.gov/eic/index.php/LEPTO

Future Updates/Additions Planned: None at this moment
Comments:

Package Name: MILOU
Purpose/Usage: MILOU is a Monte Carlo generator for deeply virtual Compton scattering (DVCS), $ep \rightarrow eY\gamma$. It is based on generalized parton distributions (GPDs) evolved to next-to-leading order.
Language(s): Fortran
Version Control: SVN
Location: /afs/rhic.bnl.gov/eic/PACKAGES/milou
Platforms: Linux, Mac OS X
Maintainers: S. Fazio, originally developed by E. Perez, L. Schoeffel and L. Favart
Documentation: MILOU: a Monte-Carlo for Deeply Virtual Compton Scattering", E. Perez, L. Schoeffel and L. Favart, hep-ph/0411389v1 and locally at https://wiki.bnl.gov/eic/index.php/MILOU
Future Updates/Additions Planned: None at this moment
Comments:

Package Name: RAPGAP
Purpose/Usage: MC generator, which can be used to generate both DIS and Diffractive e+p events
Language(s): Fortran
Version Control: http://projects.hepforge.org/rapgap
Location: http://projects.hepforge.org/rapgap/ and locally at /afs/rhic.bnl.gov/eic/PACKAGES/RAPGAP-3.2
Platforms: Linux
Maintainers: Dr. Hannes Jung at DESY
Documentation: http://projects.hepforge.org/rapgap/rapgap.pdf and https://wiki.bnl.gov/eic/index.php/RAPGAP
Future Updates/Additions Planned: Are defined by the author
Comments:

Package Name: PYTHIA
Purpose/Usage: A multipurpose generator for ep and eA scattering. The eRHIC version is based on version 6.4.13. The diffractive VM model was changed and nuclear PDFS have been integrated via LHAPDF. In addition radiative corrections have been implemented.
Language(s): Fortran
Version Control: SVN
Location: original version at http://www.hepforge.org/downloads/pythia6, the updated PYTHIA version can be found at /afs/rhic.bnl.gov/eic/PACKAGES/ PYTHIA-RAD-CORR-32BIT
Platforms: Linux
Maintainers: E.C. Aschenauer
Documentation: There is the official PYTHIA manual http://home.thep.lu.se/~torbjorn/pythia/lutp0613man2.pdf the update notes http://www.hepforge.org/downloads/pythia6 and locally https://wiki.bnl.gov/eic/index.php/PYTHIA
Future Updates/Additions Planned: update to PYTHIA version 6.4.28
Comments: PHYTHIA-8 is currently not an option, because it does not support any ep collisions

Package Name: PEPSI
Purpose/Usage: PEPSI (Polarized Electron Proton Scattering Interactions) is a Monte Carlo generator for polarized deep inelastic scattering (pDIS). It is based on the LEPTO 6.5 Monte Carlo for unpolarized DIS.
Language(s): Fortran
Version Control: SVN
Location: Locally at /afs/rhic.bnl.gov/eic/PACKAGES/PEPSI
Platforms: Linux
Maintainers: E.C. Aschenauer
Documentation: Mainly locally https://wiki.bnl.gov/eic/index.php/PEPSI
Future Updates/Additions Planned: None

Comments: This code is superseded by DJANGOH

1.1.2 Electron-Nucleus Generators

Package Name: DPMJET
Purpose/Usage: DPMJet is a multipurpose generator Based on the Dual Parton Model (DPM), developed by Stefan Roesler, Ralph Engel and Johannes Ranft. It is capable of simulating hadron-hadron, hadron-nucleus, nucleus-nucleus, photon-hadron, photon-photon and photon-nucleus interactions from a few GeV up to the highest cosmic ray energies. Our current version installed on EIC machines is modified by Nestor Armesto, taking the nuclear fission effect into consideration through FLUKA, and thus can be used in both ep and eA simulations.
Language(s): Fortran
Version Control: CVS
Location: /afs/rhic.bnl.gov/eic/PACKAGES/DPMJET
Platforms: Linux
Maintainers: Liang Zheng and J.-H. Lee
Documentation: Mainly locally https://wiki.bnl.gov/eic/index.php/DPMJet, with links to all the official documentation
Future Updates/Additions Planned: none
Comments: There have been several bug fixes (https://wiki.bnl.gov/eic/upload/A_dpmjet_use_report.ppt) implemented

Package Name: DPMJetHybrid
Purpose/Usage: DPMJetHybrid is a hybrid generator based on the multipurpose generator DPMJet with nuclear fission modified by Nestor Armesto combined with PYTHIA to simulate parton level interactions. PYTHIA and PHOJET have been employed as two available options to treat elementary interactions coordinated by a Glauber model in DPMJet. For the purpose of implementing nuclear effects on parton level, we have added an energy loss mechanism and nuclear PDFs to PYTHIA.
Although, this program has incorporated all the ability of DPMJET and PYTHIA, the major effort has been mainly devoted for the development of ep/eA simulations, thus it is advised to only use it for ep/eA at the current stage.
Language(s): Fortran
Version Control: CVS
Location: /afs/rhic.bnl.gov/eic/PACKAGES/DPMJetHybrid
Platforms: Linux
Maintainers: Liang Zheng and J.-H. Lee
Documentation: Mainly locally https://wiki.bnl.gov/eic/index.php/DpmjetHybrid with links to all the relevant documentation.
Future Updates/Additions Planned: none
Comments:

Package Name: Sartre v.1.1
Purpose/Usage: An event generator to simulate exclusive diffractive events. Sartre is applicable in ep and eA DIS at an EIC, and in pp, pA, and AA ultra-peripheral events at RHIC. Included are production of DVCS, and ρ, ϕ, and J/Ψ vector mesons. At the moment gold and protons are implemented. Sartre simulates both coherent and incoherent diffraction.
Language(s): C++ (and CMake)
Version Control: under svn at code.google.com https://sartre-mc.googlecode.com/svn/trunk
Location: Locally at /afs/rhic.bnl.gov/eic/PACKAGES/sartre
Code in form of tar-ball can be downloaded from https://code.google.com/p/sartre-mc/
Platforms: Linux, Mac OS X
Maintainers: Tobias Toll and Thomas Ullrich

Documentation: Extensive documentation (html) is distributed with the package. Also online at https://code.google.com/p/sartre-mc/
Future Updates/Additions Planned:
Adding tables for more nuclear species. Next will be lead, followed by oxygen and calcium. A major change (version 2) will be to include also inclusive diffraction.
Comments:
Sartre requires look-up tables containing the amplitudes of the process as a function of 3 kinematic variables. Producing these tables is *extremely* CPU intense. We are using the Open Science Grid (OSG). There is a user group for EIC on the OSG for this purpose.

1.2 Supported Detector Simulation Packages:

Apart from the physics generators it is essential for a project like EIC/eRHIC to have software, which allows studying the impact of detector responses on the physics observables. For this purpose two software packages have been developed both described below. One is a fast smearing generator in which detector resolutions and efficiencies are parameterized and second is a virtual MC package based on the FairRoot software tools.

Package Name: eic-smear
Purpose/Usage: Fulfills two roles:
(1) Parse ASCII events produced by event generators into C++ objects stored in a ROOT file.
(2) Approximate detector response by way of user-provided parameterizations to "smear" variables such as momentum, energy and angle. Supports the most commonly used EIC generators natively. A class hierarchy allows users to extend support to other event generators with minimal additional coding. The smearing generator is designed to be approximate, but much faster to learn and run than the full detector simulation framework.
Language: C++
Version Control: http://svn.racf.bnl.gov/svn/eic/Utilities/eic-smear
Location: /afs/rhic.bnl.gov/eic/PACKAGES/eic-smear
Platforms: Linux; Mac OS X
Maintainer: Thomas Burton
Documentation: https://wiki.bnl.gov/eic/index.php/Eic-smear [wiki.bnl.gov]
Future Updates/ Additions Planned: The code is now in a stable state. Bug fixes will be provided as encountered and native support for new generators may be added as needs arise.

Package Name: EicRoot
Purpose/Usage: A complete GEANT physics simulation environment for a dedicated eRHIC detector. Based on the FairRoot framework, developed and maintained at GSI, Darmstadt, for a number of its already-running as well as future experiments within the FAIR project. The FairRoot framework provides an easy interface to various geometry models, both the GEANT3 and GEANT4 transport engines, ROOT I/O operations, as well as modern track reconstruction tools. EicRoot intensively uses these readily available pieces of software and adapts them to the dedicated EIC detector concept. At present the EicRoot software bundle has a fairly detailed description of tracking detectors and electromagnetic calorimeters, including the minimal set of necessary simulation, digitization and reconstruction algorithms. It has an interface to the EIC smearing generator, which in particular allows one to import physics event lists and substitute a complete simulation by fast approximate response functions for the sub-detectors, which are not yet fully implemented. Timelines for the FAIR project realization are similar to the anticipated eRHIC running period, which guarantees a continuous software support from the GSI community in the future.
Language: C++
Version Control: http://svn.racf.bnl.gov/svn/eic/eicroot
Location: /eic/data/FairRoot/eicroot on the RACF EIC computing cluster
Platforms: Linux
Maintainer: Alexander Kiselev

Documentation: The distribution contains all the documentation necessary to install the software and run simulation->digitization->reconstruction chain. There exists a tutorial directory with complete examples, which can be run out of the box even by an inexperienced person.
Future Updates/Additions Planned: The package is in the active development stage. Description of new sub-detectors (in particular the PID ones), the interaction region design, more realistic and/or efficient digitization and reconstruction algorithms will be added as needed.

1.3 General User Support:

Everybody interested in doing simulations for EIC/eRHIC and who wants to use the EIC/eRHIC software framework has, like in STAR and PHENIX, to apply for a RACF account. The RACF team has set up like for STAR and PHENIX an eRHIC user group. How to apply for an RACF eRHIC account is documented in detail at the EIC wiki at https://wiki.bnl.gov/eic/index.php/Computing.
Several people at different experience levels, from undergraduate students to professors, have already successfully used the EIC/eRHIC software. One specific example is the eSTAR and ePHENIX LoI, for which the physics simulations all used the physics generators environment developed by the BNL EIC science taskforce.

2 Short and Long Term Projects:

2.1 Software environment:

In the next weeks, apart from further developing the physics generators and detector simulation tools, a common configure and MAKE environment should be set up, such that all generators are compiled with the same compiler and linker options.
In addition the analysis tools used to make all the EIC physics impact studies should be collected and archived.

2.2 EIC/eRHIC Software Preservation:

To achieve a long term EIC/eRHIC software preservation the philosophy recently established in HEP should be followed. CERN, DESY, SLAC and FermiLab all follow the strategy described in http://arxiv.org/abs/1205.4667 and at http://www.dphep.org. This would guarantee the long-term stability of the currently developed software.